REMARKS

This paper is responsive to the Final Office dated December 2, 2004. Claims 1-21 and 29-35 were examined. The Examiner has indicated that claims 4 – 6, 11 – 12, and 20 include allowable subject matter. Applicant appreciates the indication of allowable subject matter. Applicant traverses all rejections.

Rejections Under 35 U.S.C. §102(b)

Claims 1, 2, 7-9, 13-17, 29, 30 and 33-35 are rejected under 35 U.S.C. §102(b) as being anticipated by "*Managing Long Linked Lists Using Lock-Free Techniques*" by Farook (hereinafter "Farook"). Applicant traverses these rejections

As previously pointed out, Farook allows for premature reclamation, which is avoided by Applicant's claimed invention. Farook does not disclose or suggest 1) coordinating modification of reference counts; or 2) the temporal relationship between creation or destruction of a pointer and corresponding modification of a reference counter, which are some of the limitations that facilitate avoidance of premature reclamation. Applicant explained the flaws of Farook which allow for premature reclamation in the previous response, and, in this current response, emphasize some of the claimed limitations that avoid Farooks' flaws. Applicant also reasserts arguments made in the previous response regarding the temporal relationships between creating a pointer and incrementing a reference counter (as well as destroying a pointer and decrementing a reference counter), which were not addressed in the Final Rejection by the Office.

Coordination of Reference Counts and Pointer Operations Not Disclosed or Suggested by Farook

In a previous response dated July 21, 2004, Applicant stated that Farook suffered from flaws that allow for premature reclamation of a shared object. The Office's response indicates that this is not recited in the claims. However, the following limitations avoid premature reclamation and are not disclosed or suggested by Farook:

claim 1: `accessing pointers to the shared objects using lock-free pointer operations to coordinate modification of respective counts;`

claim 29: employing one or more lock-free pointer operations to maintain reference counts for one or more accessed component objects thereof; and

claim 34: means for coordinating competing access to the shared object using one or more reference counts and pointer manipulations.

The Examiner refers to Farook's cursor, TryInsert, and TryDelete procedures to support an argument that Farook discloses the above limitations. However, none of these procedures disclose or suggest these limitations. None of these procedures in Farook coordinate modification of respective counts. Neither the text of Farook nor these procedures disclose any coordination of maintaining reference counts or use of reference counts and pointer manipulations to coordinate competing access to a shared object. If there are multiple processes accessing a shared object in Farook, they modify the reference counts without coordination. Farook especially does not disclose or suggest lock-free pointer operations to coordinate modification of respective counts. Moreover, with regard to claim 34, Farook does not disclose or suggest "one or more lock-free pointer operations to ensure that if the number of pointers to the shared object is non-zero, then so too is the corresponding reference count." As stated in the previous response, Farook allows for a reference count to reach zero even if a pointer to a shared object still exists, which provides the avenue for premature reclamation.

Temporal Relationships not Disclosed or Suggested by Farook

In the previous response, Applicant explained that Farook does not disclose or suggest the limitations as recited in claim 13 regarding temporal relationships. It can be clearly seen in the cursor procedure, that a pointer is created prior to incrementing a reference counter. In cursor, a pointer is assigned as the very first operation in cursor. Afterwards, a reference counter is incremented. Claim 13 explicitly recites "access operations that, **prior to attempting creation or replication of a pointer** to any of the component shared objects, **increment a corresponding reference count**, and upon failure of the attempt, thereafter decrement the corresponding reference count." The code in Farook clearly contradicts any argument that Farook discloses or suggests incrementing a reference counter prior to attempting creation of a pointer. In addition, Farook does not even address how to modify the reference counter if

pointer creation fails. **In fact, upon *successful insertion of a node*, Farook *decrements* a reference counter, in `TryInsert`.** Claim 13 also explicitly recites "the access operations decrementing a particular reference count, except when handling a pointer creation failure, no earlier than upon destruction of a pointer to a corresponding one of the component shared objects." **In the `TryDelete` procedure of Farook, a reference count is *decremented prior to release* of the target pointer.**

With regard to claim 8, the `TryDelete` procedure in Farook relied upon by the Office Action does not recursively follow pointers of a target node that the `TryDelete` operation is attempting to delete. In the Final Rejection, the Office states the following:

> Part of the TryDelete operation is to repeatedly try to assign the next node after the target node to the previous node until this operation is a success. Therefore, the operation must recursively follow the pointer of a target node each time it tries to assign the next node to the previous node until it succeeds.

Applicant respectfully submits that the above conclusion by the Office is erroneous. Farook merely discloses calling `TryDelete` from `Delete` repeatedly upon failure of a delete attempt. A first procedure repeatedly calling a second procedure or repeatedly performing a same operation, in itself, is not recursively following pointers.

For at least the reasons discussed above, Applicant's independent claims 1, 13, 29 and 34 are allowable and not anticipated by Farook, or any other art of record. In addition, the dependent claims are allowable at least because they are dependent on corresponding ones of the above allowable independent claims.

Rejections Under 35 U.S.C. §103

Claims 3 and 21 are rejected under 35 U.S.C. §103(a) as being unpatentable over Farook. Claim 10 is rejected under 35 U.S.C. §103(a) as being unpatentable over Farook in view of "*Simple, Fast and Practical Non-Blocking and Blocking concurrent Queue Algorithms*" by Michale (hereinafter "Michael"). Claims 18 and 19 are rejected under 35 U.S.C. §103(a) as

being unpatentable over Farook in view of "*Transactional Memory: Architectural Support for Lock-Free Data Structures*" by Herlihy (hereinafter "Herlihy"). Claims 31 and 32 are rejected under 35 U.S.C. §103(a) as being unpatentable over Farook in view of "*Garbage Collection: Algorithms for Automatic Dynamic Memory Management*" by Jones (hereinafter "Jones"). Applicant respectfully traverses all of these rejections.

All of the rejections under 35 U.S.C. §103 rely on Farook. The above discussion and the prior response already states the failure of Farook to disclose or suggest any of Applicant's claims. Farook, Michael, Herlihy, and Jones, standing alone or in combination, do not disclose or suggest any of Applicant's claims. Claims 3, 10, 18, 19, 21, 31, and 32 depend on one of the above allowable independent claims and are allowable at least for the reasons already discussed.

Conclusion

In summary, claims 1-21 and 29-35 are in the case. All claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that, on the date shown below, this correspondence is being

☐ deposited with the US Postal Service with sufficient postage as first class mail, in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

☒ facsimile transmitted to the US Patent and Trademark Office.

Steven R. Gilliam — 2-Feb-2005 (Date)

EXPRESS MAIL LABEL: _____

Respectfully submitted,

Steven R. Gilliam, Reg. No. 51,734
Attorney for Applicant(s)
(512) 338-6320
(512) 338-6301 (fax)